A process variant modeling method comparison

Aysolmaz, Banu; Yaldiz, Ali; Reijers, Hajo

published in
Enterprise, Business-Process and Information Systems Modeling - 17th International Conference, BPMDS 2016, 21st International Conference, EMMSAD 2016, Held at CAiSE 2016, Proceedings

2016

DOI (link to publisher)
10.1007/978-3-319-39429-9_18

document version
Publisher's PDF, also known as Version of record

A Process Variant Modeling Method Comparison: Experience Report

Banu Aysolmaz[1(✉)], Ali Yaldiz[2], and Hajo Reijers[1]

[1] Vrije Universiteit Amsterdam,
De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands
{b.e.aysolmaz,h.a.reijers}@vu.nl
[2] 4S Information Technologies, Ankara, Turkey
ali.yaldiz@4s.com.tr

Abstract. Various process variant modeling methods have been introduced in the literature to manage process diversity in a business context. In industrial settings, it is difficult to select a method suitable for the needs and limitations of the organization due to the limited number of examples and guidelines. In this paper, we report our experiences on variant modeling in a process management consultancy company. The company experienced difficulties in maintaining and reusing process definitions of their customers and decided to evaluate variant modeling methods as a solution. We selected two methods, the Decomposition Driven and the Provop, to develop variant models of seven software project management processes from five customers. We evaluated the results together with company experts. This study contributes to the field by providing real-life examples of two variant modeling methods, a comparison of the results with these methods and a guideline for choosing a method under comparable conditions.

Keywords: Business process modeling · Process variant modeling · Decomposition driven method · Provop

1 Introduction

In enterprises, business process modeling (or process modeling for short) is of great importance to reveal processes and develop business process management systems (BPMS). In process modeling, one of the problems that analysts encounter is the need to deal with process variability. Due to the diversity in business contexts, variants of the same process may be modelled and used in multiple cases in the same organization [1]. This diversity may be caused by various factors such as differences in delivered products, customer types, and divergent business requirements in countries. When such factors are present, consideration of process variants during process modeling is inevitable [2]. However, in the design of a process model, it is a challenging task to either maintain variants of the same process separately while managing the relations between them or integrate the process variants into a single model while preventing complexity and redundancy [2].

R. Schmidt et al. (Eds.): BPMDS/EMMSAD 2016, LNBIP 248, pp. 285–300, 2016.
DOI: 10.1007/978-3-319-39429-9_18

To overcome such difficulties, various methods have been proposed to incorporate variant management into the phases of the business process management (BPM) life cycle [3]. These methods provide solutions for different cases. However, in real-life settings, it is difficult for an organization to make the proper choice between variant modeling methods. It is hard to find studies from the literature on the evaluation and comparison of methods in practice as well as any guidelines to make a method selection [3]. The study presented in this paper stems from the difficulties observed within 4S Information Technologies (4S for short), a company that provides consultancy services to its customers to analyze and improve their processes and develop BPMSs using HP PPM tools [4]. For each customer, 4S defines a new variant of a process, such as software project management, demand management, software change request management, risk and issue management, etc. 4S maintains separate process definitions and artefacts for each variant, yet the interrelations between the variants are not tracked. As a result, 4S cannot systematically reuse its process knowledge for creating a new variant for a new customer. The same problem applies for maintenance, as they need to update each variant independently without the opportunity to reuse the effort. For these reasons, 4S was motivated to implement a variant management method to more efficiently apply its knowledge in process analysis, design and improvement activities.

In accordance with the needs of 4S, the aim of this study is to implement and compare process variant modeling methods in a real-life setting. For this purpose, we selected two different, well-accepted process variant modeling methods focusing on the analysis and design of process variants: the Decomposition Driven method [5] and the Provop Method [1]. We applied these methods to 7 software project management processes of 5 4S customers. A team of 6 employees from the company participated in the study that was led by one of the authors. The team evaluated the application of the two methods in terms of effort spent, structure and flexibility of the outputs for maintenance and utilization in new projects. On the basis of this evaluation, we present a guideline that companies may follow when they face a similar situation.

The rest of the paper is structured as follows. Section 2 presents the design of this study describing the organization, the need, method selection, the purpose and the plan. In Sect. 3, we explain how we applied the Decomposition Driven method and in Sect. 4, the Provop method. Section 5 includes an evaluation of the results together with guidelines for method selection. Section 6 concludes the study.

2 The Design of the Process Variant Modeling Study

2.1 The Organization and the Need for Variant Modeling

4S is a consultancy company that provides process analysis, improvement and automation services to its customers using HP PPM product [4]. HP PPM provides a flexible workflow development environment specializing in project and demand management processes. 4S has customers from various countries and industries focusing on different process areas. Usually, 4S analysts need to rely on their own expertise to discover other activities and improve the existing process. They cannot systematically

exploit process knowledge obtained from previous similar companies for new customers. Based on the problem, the need for using a process variant modeling method for 4S can be summarized as follows:

- When they start to work with a new customer, 4S analysts need to combine their knowledge on previous customers as a baseline for understanding the new as-is process and suggesting improvements. Analysts would be better off if they would have an integrated model, which they can practically use as a jump start in the project initiation phase.
- Through the steps of process analysis, improvement and design, 4S analysts design various processes for customers. Even when they start developing a process based on a previously encountered process, the knowledge of such related processes and the connections hereto are soon lost. Analysts cannot benefit from one another's experiences as it is hard for them to go over each process to find out if it is relevant for a new case. The same problem persists through process enactment phase; as developers cannot easily find out similar automated processes and activities for example, to reuse their form design and flow logic.
- When an improvement or update is needed, 4S needs to go over each customer's processes to find out which ones are affected and where updates are needed. This requires a lot of effort and can introduce errors due to manual review process.

2.2 Process Variant Modeling Method Selection

4S needs a process variant modeling method to manage customer models in an integrated way, utilize the knowledge in the following projects and enhance maintainability of multiple process models. Process variant modelling approaches have been proposed in the literature over a spectrum of *single* to *multi* model solutions [5]. On the one end of the spectrum, multi-model approaches capture every possible variant of a process as a separate model. Using such an approach, redundancy and maintainability problems are introduced, which is basically the problem that comes with not managing process variants at all [6]. On the other end of the spectrum, methods that model all variants in an integrated single model produce integrated models for multiple variants. The resulting decrease in the total number of process elements and the improvement of maintainability is balanced against increased complexity and comprehension problems. As a result of benefits, single model approaches are more popular in the industry. Considering the situation at hand, we focused on single-model approaches as well.

Single model approaches apply different techniques to integrate multiple process variants into a single source and use the single source to configure a specific process variant. Such techniques include questionnaire-based models, feature models, goal models, and decision tables [3]. Some single model approaches decrease complexity via providing only delete and condition selection operations based on a comprehensive base model. In 4S case, best practice model developed based on PMBOK guide was used as the starting point as it is common for various domains [3]. PMBOK is a book that provides a set of guidelines to define and implement project management processes such as scope, time and quality management. However, the best practice model of 4S is

not inclusive of all activities – it is rather a brief process model including must-have activities. For this reason, we needed a method that has more flexibility to define process variants. Another criteria for method selection was on the need of a variant modeling tool. For some approaches, a tool that has specific process variant modeling features are required to properly benefit from the approach [7]. Due to the concern of increase in effort by adding a new tool to company repository and training needs, 4S eliminated the methods that need a specific tool.

Considering these needs and limitations, we identified an initial list of process variant modeling methods based on existing literature reviews [3] and our review of the related work. We discussed potential pros and cons of these methods with the 4S team and made a joint selection. As stated, we selected the Decomposition Driven method and the Provop method.

The Decomposition Driven method was selected because it provides flexibility for certain parts of the model. By means of step-wise decomposition, users can choose to model some of the sub-processes together and some others separately [5]. Moreover, the team specifically appreciated how the method does not only approach identification of variants mechanically but it considers the wider business environment via business drivers and syntactic drivers inherent to the processes. In turn, the Provop method was selected due to its robust mechanism to treat all variants equally and create a big model. The usage of the list of options to mechanically end up in new variants in a plug and play logo-like feeling was seen as another advantage. The team focused on selecting methods that have a different approach for variant modeling. In this way, a comparison of the benefits of different process variant modeling approaches for future use would be feasible.

2.3 Purpose of the Study

Based on the identified problems and the needs of 4S, we formulated the following questions:

- How can we develop process variant models for a process where different process models are developed for diverse companies although they share the same best practices?
- How does the application of two variant modeling methods, the Decomposition Driven and the Provop methods, compare for flexibility in terms of reusing the knowledge to define processes for new customers and maintain all variants in case of a change in one process?
- What factors are to be considered for an organization to select a proper variant modeling method based on its setting, needs and constraints?

2.4 Process Variant Modeling Plan

4S decided to use software project management processes of five customers for this study. Four of the companies are from Turkey and other is a Turkish branch of an

international company. For all of them, their software project management processes are defined based on PMI's PMBOK guide [8]. One of the authors of this paper who is affiliated with 4S, worked as a leader of the team in the company that implements the methods and evaluates the results.

Although 4S uses the PMBOK guide as the baseline, the best practices provided just the essential steps of a project management process. The processes were defined as workflow definitions on HP PPM, but process models were not developed for analysis purposes. We converted the low level workflow models to process models in BPMN notation through discussion sessions with the experts. We aggregated workflow tasks to higher level activities in BPMN. The experts found it easier to observe and define relations between process variants using the BPMN models. For this reason, we decided to use these models in variant modeling activities. For each variant, we developed a high level software project management process. We created a relation table for the corresponding workflow tasks for each BPMN activity. In this way, we achieved more comprehensible process definitions where the experts could better observe the relations between the variants. Still, we are able to map process model activities to workflow tasks via the relation table. This enabled the experts to analyze workflow definitions together with variant models after the study is completed. A summary of the companies and their process metrics can be found in Table 1. Process models for all process variants can be seen in [9].

Table 1. Metrics for software project management processes of 4S customers

Process	Field	Number of workflow tasks	Number of workflow gateways	Number of BPMN activities	Number of BPMN gateways
Company 1	Annuity Insurance	15	7	10	4
Company 2	Insurance	40	9	12	6
Company 3	Banking	21	14	9	0
Company 4-1	Banking	48	7	14	6
Company 4-2	Banking	8	2	9	2
Company 5-1	Telecom	46	11	11	2
Company 5-2	Telecom	44	8	11	2
Average		31.7	8.3	10.9	3
PMBOK best practice		13	0		

The described research project was initiated based on the need in 4S as defined in previous sections. After the analysis of related work, elimination took place of multi-model approaches and approaches that require usage of a specific variant modeling tool and that conduct automated process discovery. Subsequently, the approaches mentioned earlier were selected. Upon the selection of the methods, the following steps were planned:

- Identify the process to apply selected process variant modeling methods: The team selected the software project management process, which is the most frequent process that they provide consultancy for their customers.
- Identify the context for application: Five customers were identified that are representative for different industries. Two of the customers implement two variants of software project management process.
- Define process models for each variant: We developed process models in BPMN notation for each customer as described at the beginning of this section.
- Apply the Decomposition Driven and the Provop methods to develop process variant models: The team conducted the relevant steps for applying the two methods as described in the following sections. Two methods were applied in parallel to prevent the effect of the learning curve.
- Evaluate the process of method application and outputs: The team collected data on the effort spent on each method and compared the outputs. In addition to comparing the outputs and facts, we conducted interviews to understand how the experts interpret the usability, complexity and efficiency.
- List the guidelines to choose proper method: The team identified the benefits and disadvantages brought by the two methods and how can one select the proper method with respect to priorities and benefits expected.

3 Applying the Decomposition Driven Method

The method starts with the definition of a main top-level process [5]. Then, each activity in the main process is defined in detail in a sub-process. Later, the sub-processes is further decomposed into sub-processes until there is no meaningful decomposition possible. At every level, the so-called variation map is created which contains activities and relations necessary to configure every variant. In the following sections, we describe the conduct of each step as prescribed by the method [5].

3.1 Step 1 – Model the Main Process

We started to apply the method by developing a main software project management process that acts as a process map applicable for all variants. The high level process can be seen in Fig. 1. Only one activity, “Plan Resources” was added and the remaining activities were directly used from the best practice. While modeling the main process, we also investigated and summarized each company’s existing processes in order to point out how they add value to the process.

3.2 Step 2 – Identify Variation Drivers

An outstanding feature of the Decomposition Driven method is the consideration of business and syntactic drivers to understand the emergence of variations and using

them to flexibly develop the models [5]. Business drivers are determined based on factors such as: resources used, products and services produced, customers, countries. In our case, we focused on how the high level activities in the main process are performed and possible causes of variation. We observed that the main cause of variation is the variety of customers. Another driver is identified as location of the services. This driver is used to differentiate the processes of Company 4, leading to variations of national and international services.

Syntactic drivers are the second type of drivers which diversify the way multiple variants produce their outcomes. They are defined based on the similarity of the process models of the variants. The method allows consolidation or separation of variant models due to syntactic drivers. In 4S, we manually assessed the similarity of process variant models with respect to the main process modeled in Fig. 1 [9]. We conclude that there is no explicit syntactic driver, as the main process can be used to reach the variant models by mostly adding nodes and alternative paths to the main process.

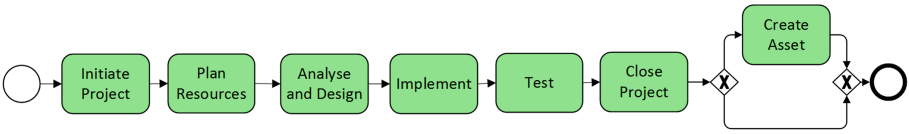

Fig. 1. Software project management high level process

3.3 Step 3 – Assess the Relative Strength of Variation Drivers

In this step, variation drivers are analyzed to specify their priority as well as their effect on defining variants. The business driver with high priority is the variety of customers in our case. Additionally, we have the driver for location of services. This driver is used to define variations for Customer 4: International vs. national services. Although Company 5 had two processes, only one variant is specified as there is no applicable, distinguishing driver.

3.4 Step 4 – Identify the Variations in Each Subprocess

In this step, we populated a variation matrix for each subprocess of the main process, as seen in Table 2. To generate this matrix for every activity in the high level process, we identified subprocesses for each driver. We then named each different subprocess and marked the subprocesses used by every driver in the matrix. For example in Table 2, *Simple Initiation* variant of *Initiate Project* subprocess is used by Company 3 and Company 4-2. This subprocess includes simple project definition activities on the system. *Complex Initiation* subprocess used by Company 1 includes a wider extent of activities such as approval of scope, project manager assignment and quality control initiation. The activities in subprocesses and all similarity decisions can be seen in [9].

Table 2. Variation matrix showing varying activities of first level subprocesses

	Initiate project	Plan resources	Analyze & design	Implement	Test	Close project	Create asset
Company 1	Complex Initiation	Moderate Planning	Basic Analyze and Design	Basic Implementation	Basic Test	Detailed Closure	
Company 2	Moderate Initiation	Complex Planning	Detailed Analyze and Design	Detailed Implementation	Detailed Test	Complex Closure	Asset Creation
Company 3	Simple Initiation	Basic Planning	Basic Analyze and Design	Basic Implementation	Basic Test	Basic Closure	Asset Creation
Company 4-1	Detailed Initiation	Simple Planning	Detailed Analyze and Design	Detailed Implementation	Detailed Test	Fast Closure	
Company 4-2	Simple Initiation	Fast Planning	Basic Analyze and Design	Basic Implementation	Basic Test	Simple Closure	Asset Creation
Company 5	Basic Initiation	Detailed Planning	Detailed Analyze and Design	Detailed Implementation	Detailed UAT	Moderate Closure	

3.5 Step 5 – Perform Similarity Assessment of Variants for Each Subprocess

In this step, we performed a similarity assessment by analyzing each subprocess of the variation matrix in Table 2. We asked the experts to identify the similarity of activities in the subprocesses for each driver. To evaluate the similarity, the experts focused on how those activities are performed. For this, they investigated the information on data used and produced while performing activities, the number of workflow steps, and the role of performer to investigate the similarities between subprocesses. As a result, activities in different subprocesses that have high similarity were marked. For example, *initiation approval* activity in *Moderate Initiation* variant was indicated to have high similarity with *initiation announcement* of *Detailed Initiation* variant.

3.6 Step 6 – Construct the Variation Map

As outputs from Step 4 and Step 5, we have the variants of subprocesses for each activity in the high level process and a list of similar activities in the subprocesses. We mapped these variants in the variation map as seen in Fig. 2. We used the decision framework of the Decomposition Driven method to decide merging of activities in the

variant map [5]. For example, *Moderate Initiation* and *Detailed Initiation* variants were merged as they were assessed to be similar. For both *Plan Resources* and *Close Project* activities, a different subprocess was defined for each variant. Only two variants among six were assessed to be similar for both activities. Rest of the subprocesses had very strong drivers and were assessed to be not similar. Thus, there were five variations of these activities as seen on the variation map. The details on activities in the subprocesses, the similarity decisions for activities and the merging of subprocesses can be seen in [9].

3.7 Step 7 – Configure a Specific Process Variant

The generated variation map acts as a reference model to observe both the process map and help the experts to arrive at possible variations by means of the flow defined by gateways. This model does not include knowledge of a specific variant. Thus, if one wants to configure a process variant, she needs to understand that specific variant and go through the variation map to select relevant activities. This selection is done for Company 4 as shown with darker colored activities in Fig. 2. We manually verified that we can generate all our variants as syntactically correct and sound.

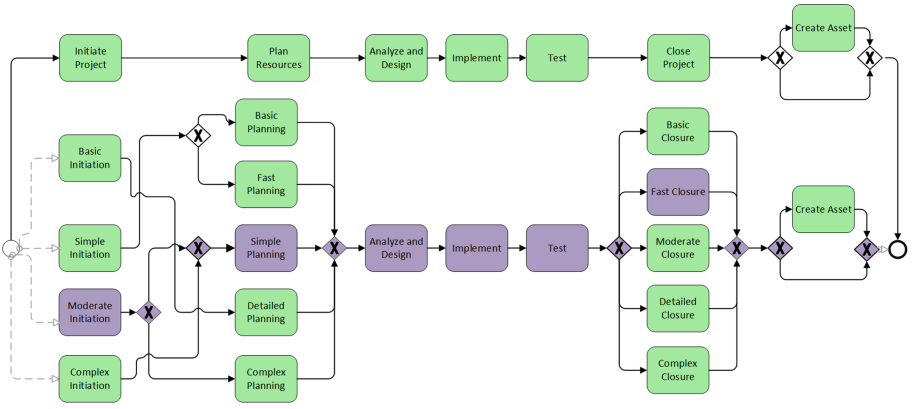

Fig. 2. Variation map for software project management process

After this step, the Decomposition Driven method suggests the iteration of all steps for the subprocesses of the main process. We applied the Decomposition Driven method completely in the first level of decomposition in 4S. Moreover, we identified the activities to be placed in each subprocess and discussed a sketch of the variation maps with the experts. In this way, the experts were able to observe how the Decomposition Driven method provided a flexible way of variant modeling in different granularity levels. For example. For "Implement" process, variation in the high level is not found necessary. However, it is observed that variants of this subprocess need to be handled considering other business drivers such as project type.

4 Applying the Provop Method

The Provop method focuses on creating a single base process model which includes adjustment points and their related sets of options [6]. The options include a set of atomic operations such as insert, delete, move and modify; which are used to configure the base model to reach a certain variant. Defining the set of operations options provides a reusable mechanism to define common operations for multiple variants. This mechanism decreases the complexity and increases controllability to configure a variant. Moreover, the Provop method can support automated variant configuration by defining context-aware configuration options.

4.1 Step 1 – Design a Base Process

The Provop method offers different policies to identify the base process on which the process variants are configured. One can either use the standard reference process used within the particular industry, use the most frequent process variant, design a version that has minimal average distance to all variants, or create a superset or intersection of all process variants. In our case, we applied a combination of these policies. First, the standard PMBOK reference model is taken as the starting point. Next, we extended this model by consideration of the policy 2, that is the variant of Company 1 which is the most frequent process worked on in 4S. We utilized policy 3 to identify process elements so that it will require the least number of operations in total to reach process variants while we also included activities at the intersection of all variants as suggested by policy 5. As a result, the base model evolved from the initial best practice model in Fig. 3 to final version in Fig. 4. Here we can indicate that it was relatively easy for us to design the base process, as we already know variant processes beforehand and we had a relatively simple and linear high level process.

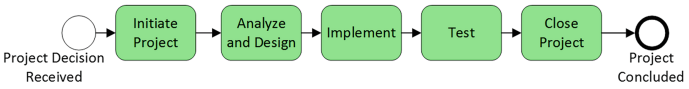

Fig. 3. Best practice process model

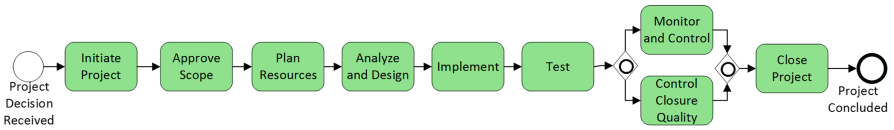

Fig. 4. Final base process model

4.2 Step 2 – Define Adjustment Points

The next action is to determine the explicit positions of the adjustment points that specify where the options can be applied on the base model. In this step, we analyzed the base process model and identified the adjustment points necessary to be able to generate all process variants provided in [9]. The final model with adjustment points can be seen in Fig. 5.

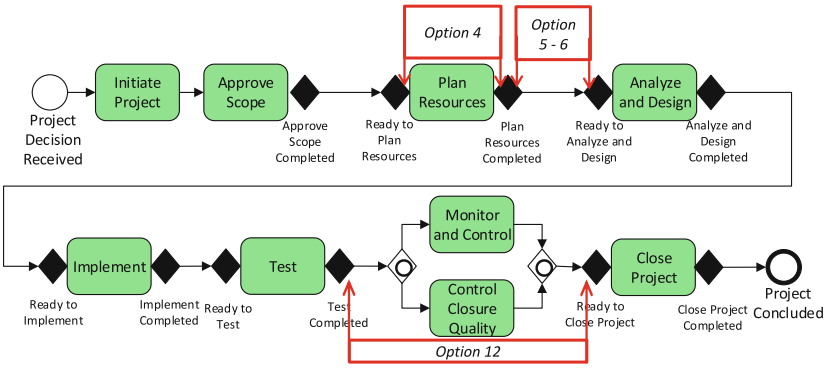

Fig. 5. Base process model with adjustment points (Color figure online)

4.3 Step 3 – Design and Model the Options

In this step, options of the process should be designed and modelled. To this end, the possible change operations for generating the variants based on the base process are investigated. Then, the conditional branches in the model are examined in order to determine that they are only variant-specific or included in all variant models. Granularity of options, the number of operations combined to define options, is important to enhance reusability and maintainability of options while keeping the number of options minimal [1]. As an example we can consider Option 1 (that contains the operation of "Delete Approve Scope") and Option 7 (that contains the operation of "Insert Control Plan Quality") seen in Fig. 6. To configure company 3 variant, we could create an integrated option including the operations in Option 1 and 7. However, in that case, we would need to define another option for the operation of "Delete Approve Scope" for company 5 variant. This would increase the number of operations and decrease reusability and understandability. Therefore, we divided this option into two as Option 1 and Option 7. 14 options are identified in total, which include 17 operations. An example set of operations are shown in Fig. 6.

4.4 Step 4 – Configure Variants

For variant configuration, the Provop suggests the usage of three substeps. First, relevant options need to be selected to configure the relevant process variant. This can be

- **Option 4:**

Context Rules: Company 2

- MODIFY Plan Resources as parallel activity
  - Gateway = Parallel gateway
- INSERT Prepare WBS parallel with Plan Resources
  - *From Point:* Ready to Plan Resources
  - *To Point:* Plan Resources Completed

- **Option 6:**

Context Rules: Company 1, 2, 4

- INSERT Control Plan Quality
  - *From Point:* Plan Resources Completed
  - *To Point:* Ready to Analyze and Design
- MODIFY Approve Plan as optional activity
  - Gateway = Exclusive gateway

- **Option 5:**

Context Rules: Company 1, 2, 4-1, 5

- INSERT Approve Plan
  - *From Point:* Plan Resources Completed
  - *To Point:* Ready to Analyze and Design

- **Option 12:**

Context Rules: Company 1, 2, 4-1, 4-2

- MODIFY Gateways = Parallel gateway
  - From Point: Test Completed
  - To Point: Ready to Close Project

Fig. 6. Example set of options for company 2 variant

done by asking users to manually choose specific variants, which is hard if there are a lot of options and specialized knowledge is required. To overcome the problem, the Provop suggests the definition of context rules by identifying, for each option, the context in which the options are applicable. In our case, the available knowledge on business drivers became useful to define the context. For each option, we identified the set of variants that are to be configured via this option. This can be seen in Fig. 6 as context rules.

Another point to be considered while applying the options is the possible constraints with the options. For example, there may be implication relation between options, an option implying the usage of another one [1]. We had an order constraint for options 5 and 6, as option 5 always needs to be applied before 6. We observed that the modelers need to pay special attention for constraints especially for options effective on the same adjustment point pairs.

In conformance with the constraints, we manually apply the set of options shown in Fig. 6 to the base process as indicated with red markers on Fig. 5 to achieve the variant process of company 2 as in Fig. 7. In the following section, we evaluate the results of applying the two methods and provide a guideline for selection.

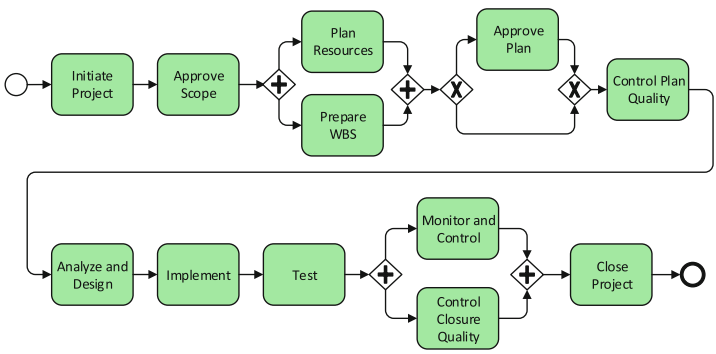

Fig. 7. Company 2 process model after configuration

After the definition of the base process model with adjustment points for the software project management process, we needed to analyze variants for subprocesses of the activities in the base process model. We were not able to identify specific guidelines for applying the Provop method in a hierarchical process structure. We plan to define subprocesses for each variant and conduct the same set of steps to develop base process models of each subprocess. However, we need to consider that new activities may be added to the high level process via options. In this case, we plan to define base process maps for the subprocesses under those activities as well. This will introduce problems in reading, as the user is not able to see and associate such a subprocess in the base process model. We also observed that special attention is needed to prevent conflicts among options for different levels of granularity.

5 Evaluating the Results

As 4S is evaluating two variant modeling methods to implement in all its projects in the future, it is important to identify the method that is practical to apply and meets the needs for reusing process knowledge and maintaining multiple variants. For practical reasons, we evaluated the effort spent on applying the two methods, structure of the outputs and flexibility of using the outputs in new projects and maintaining them when there is an update in one of the variants.

22 h were spent in 5 sessions for the Decomposition Driven method, whereas 15 h were spent in 4 sessions with the Provop method. The experts appreciated the idea of incorporating the business context to identify sources of variation. However, the variety of customers was already an explicit business driver for 4S from the start of the study. The experts think that their extra effort for the Decomposition Driven method will pay off when they implement the method for low level subprocesses and other process types with potentially more varied business drivers.

Comparing the structure of the resultant models, variation map of the Decomposition Driven method has 25 activities, 10 gateways and 50 edges. The Provop method produced a simpler model with only 9 activities, 2 gateways, 13 edges and 11 adjustment points as customized elements. The Decomposition Driven method seems to produce a bigger and more complex model (due to edges/activities ratio). However there is an extra artefact, list of options, required to read and customize the Provop base process model. The experts indicated that it was easier for them to read the Provop's base process model and "picturize" how the adjustments may be conducted even without seeing the option list. They found it non-intuitive to interpret the variation map of the Decomposition Driven method, e.g. in particular with respect to finding out where to start reading the process and how to configure a specific variant. This point reduces the flexibility to maintain existing variants. On the other hand, the experts found it more flexible to use the variation map for defining a new process, as they can see all options together with constraints on the map.

The experts appreciated the flexibility of the Decomposition Driven method for modeling variants in different granularity levels. It is conventional to develop separate models when there are variants of subprocesses which are very different from each other although the higher level process is similar. In this way, it is possible to balance

complexity and comprehensibility on multiple process models. For example, no variation was needed at this level for "Initiate Project" and "Close Project" activities. The variations in lower level activities are to be evaluated in variant modeling of their subprocesses.

The 4S experts found it easier to use the Provop base model for configuration of process variants. Similar change operations grouped in the Provop decreased the complexity to generate a variant and made it easier to configure a variant without much knowledge of the customer. Variation map of the Decomposition Driven method does not provide any information on variants, one needs to have specialized knowledge. Lastly, in case organizations need automated configuration of variants, the Provop method provides functionality to integrate variant management with some modeling tools [10]. In the following section, we provide a list of guidelines to help organizations to choose the relevant method for their setting, needs and constraints.

5.1 Guidelines for Process Variant Modeling Method Selection

Based on our evaluation described in the previous sections and the feedback from the 4S experts, we identified the guidelines in Table 3 to make a selection between the two methods.

Table 3. Variant modeling method selection guidelines

Needs and constraints of the organization	Suggested method
You want all information to be embedded in your models and your main purpose for using variant models is to reuse knowledge when you need to create a new variant	Decomposition Driven Method
You need a stepwise description of the operations to configure all your existing variants	Provop Method
You cannot use extra constructs in your modeling tool	Decomposition Driven Method
You plan to use automated process variant configuration	Provop Method
You need a hierarchical approach to analyze your process variants and you have different levels of variability in your subprocesses	Decomposition Driven Method
You prefer a lean model where your domain experts can easily understand the base model and your analysts can go deeper in configuring variants using extra information	Provop Method

We observe that the experts can benefit from the merits of the two methods even when they use another method specifically in the following ways:

- The guidelines of the Decomposition Driven method may be used to extensively evaluate why process variants emerge in your business context.
- The approach of the Decomposition Driven method for modeling variants of hierarchical process models may be implemented in other methods as well.

- The policies of the Provop method to define a base model, such as usage of reference models, most frequent variant or minimal distance, may be used while defining a main process in any single-model variant modeling approach.
- When the generated single-model does not include the steps of configuring a specific process variant, option list approach can be used.

6 Conclusion

In this study, we implemented two different single-model approach process variant modeling methods in a real-life setting. The involved company has various process definitions of the same process as they provide BPM services in similar areas to their customers. The company experienced problems in reusing their existing knowledge to define a new process for a customer, and finding out the related process variants and maintaining them properly when there is an update in an existing process. To explore the solutions to their problems, they wanted to employ process variant modeling methods to evaluate their benefits and compare with each other. For this purpose, we selected the Decomposition Driven and the Provop methods. We observed that both of these methods can be applied with a reasonable effort and will bring benefits by providing a single integrated model to configure models. We observed benefits of both methods in 4S from different aspects.

As is the case for many things in life, there is not a single answer for the question which method to select. Both methods we analyzed here have their merits while they still introduce complexity due to new analysis techniques and notations to be applied. Even when professionals decide to use another variant modeling method or no method at all, learning about variant analysis through these methods will bring benefits. For example, when organizations explore business drivers causing variations, they can use this information to evaluate root-causes and deal with this variation on a strategical level. Another point is that every organization can adopt the idea of using policies to define its base models. Considering this fact, we prepared a list of guidelines to help organizations to select a proper method and to utilize the insights provided by these methods when they have process variation in their organization.

In future work, we will completely apply the methods for low level processes of software project management as already initiated in current work. This will enable a thorough evaluation of the methods for hierarchical processes. Also, we plan to apply the methods to the demand request process, which even shows more variation with respect to customers and other factors. In parallel, 4S plans to start a gradual usage of variant modeling in its company. For this, new experts will be trained. Then, prototypes will be identified from the projects where the experts will use the outputs of this study to define processes of the new customers.

Acknowledgements. This study has been conducted in 4S Information Systems in collaboration with process management team of the company. This study received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 660646.

References

1. Hallerbach, A., Bauer, T., Reichert, M.: Capturing variability in business process models: the Provop approach. J. Softw. Maint. Evol. Res. Pract. **22**, 519–546 (2010)
2. Döhring, M., Reijers, H., Smirnov, S.: Configuration vs. adaptation for business process variant maintenance: an empirical study. Inf. Syst. **39**, 108–133 (2014)
3. Ayora, C., Torres, V., Weber, B., Reichert, M., Pelechano, V.: VIVACE: a framework for the systematic evaluation of variability support in process-aware information systems. Inf. Softw. Technol. **57**, 248–276 (2015)
4. Enterprise, H.P.: Project and Portfolio Management – PPM. http://www8.hp.com/us/en/software-solutions/ppm-it-project-portfolio-management/
5. Milani, F., Dumas, M., Ahmed, N., Matulevicius, R.: Modelling families of business process variants: a decomposition driven method. CoRR abs/1311.1 (2013)
6. Hallerbach, A., Bauer, T., Reichert, M.: Configuration and management of process variants. In: Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 1 SE - 11, pp. 237–255. Springer, Heidelberg (2010)
7. Conforti, R., Dumas, M., Rosa, M.La., Maaradji, A., Nguyen, H.H., Ostovar, A., Raboczi, S.: Analysis of Business Process Variants in Apromore (2015)
8. Project Management Institute Inc: A guide to the project management body of knowledge (PMBOK® guide) (2000)
9. Yaldiz, A.: Evaluation of process variant modeling approaches: a case study, Ankara, Turkey (2016). http://expertjudgment.com/publications/METU_II_TR_2016_YILDIZ.pdf
10. Reichert, M., Rechtenbach, S., Hallerbach, A., Bauer, T.: Extending a business process modeling tool with process configuration facilities: the Provop demonstrator. In: Proceedings of BPM 2009 Demonstration Track (2009)